

# TP2 (version JavaScript)

---

INF8808: Visualisation de données

Département de génie informatique et génie logiciel



# POLYTECHNIQUE MONTRÉAL

**Auteure:** Olivia Gélinas

**Chargé de lab:** Hellen

## Objectifs

---

L'objectif de ce travail pratique est de créer un diagramme à bandes groupées interactif à l'aide de données ouvertes en format CSV.

Avant de commencer, nous vous recommandons d'avoir effectué les lectures et exercices suivants:

**Lectures:** Chapitres 6, 7 et 8 du livre de Scott Murray

---

- Chapitre 3 - 1, 2, 3

**Exercices:**

- Chapitre 4 - 1, 2
- Chapitre 5 - 1, 2

## Introduction

---

Un diagramme à bandes est un type de visualisation de données qui représente des données à l'aide de rectangles dont les longueurs sont proportionnelles aux valeurs qu'ils représentent. Dans un diagramme à bandes groupées, les bandes sont placées en groupes en utilisant leur position sur le graphique. Dans chaque groupe, la couleur de chaque bande indique son sous-groupe. Les graphiques à bandes groupées sont couramment utilisés pour comparer les valeurs de chaque sous-groupe à travers les différents groupes.

Dans ce travail pratique, vous allez implémenter un diagramme à bandes groupées à l'aide des données de la pièce de Shakespeare, *Roméo et Juliette*. Les données ont été extraites de Kaggle, un site Web populaire au sujet de la science des données [1]. L'ensemble de données a été modifié pour obtenir la version à utiliser dans ce laboratoire. Il contient des données sur texte entier de *Roméo et Juliette*.

## Description

---

Dans ce travail pratique, vous devrez compléter le code JavaScript fourni en utilisant la librairie D3. Le but sera d'afficher un diagramme à bandes groupées avec des données sur le nombre de lignes prononcées par chaque joueur dans chaque acte de la pièce. Pour rendre le graphique interactif, vous implémenterez également le code pour afficher une info-bulle informative lorsque chaque bande est survolée par le curseur. De plus, il y aura une légende indiquant la couleur correspondant à chaque joueur affiché sur le graphique.

Chaque groupe contiendra comme sous-groupes une bande pour chacun des 5 joueurs qui prononcent le plus de lignes dans la pièce. Les autres joueurs seront regroupés dans leur propre catégorie « *Other* », contenant leur total de lignes.

Les sous-sections suivantes présentent les différentes parties que vous aurez à compléter pour ce laboratoire. Pendant que vous codez, nous vous recommandons de compléter le prétraitement des données d'abord, suivi de la mise en œuvre du diagramme à bandes lui-même. Les deux parties suivantes, la légende et l'info-bulle, sont indépendantes l'une de l'autre.

## Structure des fichiers

Pour accomplir ce travail, vous devrez remplir les différentes sections **TODO** dans les fichiers de l'archive fournie pour le laboratoire. Les commentaires dans le code expliquent en plus de détails les étapes à suivre.

Dans ce travail pratique, nous vous fournissons une archive contenant 6 fichiers JavaScript utilisés pour accomplir la visualisation souhaitée :

- **index.js**: Ce fichier représente le point d'entrée du code et orchestre les différentes étapes nécessaires à la réalisation de la visualisation. Il n'a pas besoin d'être modifié.
- **scripts/helper.js**: Ce fichier contient quelques fonctions de base nécessaires pour afficher la visualisation. Il n'a pas besoin d'être modifié.
- **scripts/legend.js**
- **scripts/preprocess.js**
- **scripts/tooltip.js**
- **scripts/viz.js**

## Données

L'ensemble de données se trouve dans le répertoire **src/assets/data** dans l'archive fournie pour le laboratoire. L'ensemble de données contient les colonnes suivantes :

- **Act**: Cette colonne représente l'acte dans lequel la ligne est prononcée.
- **Scene**: Cette colonne représente la scène dans laquelle la ligne est prononcée.
- **Line**: Pour une  $n$ -ième ligne dans une scène donnée, cette colonne contiendra la valeur  $n$ .
- **Player**: Cette colonne contient le nom du joueur qui a prononcé la ligne.
- **PlayerLine**: Cette colonne contient le contenu prononcé par le joueur pour cette ligne.

**Remarque:** Comme rappel, le jeu de données représente les données d'une pièce de théâtre, qui sont généralement divisés en gros morceaux appelés « actes ». Dans cette pièce, il y a 5 actes. Puis, chaque acte est divisé en scènes qui contiennent des lignes présentées dans l'ordre dans lequel elles sont prononcées pendant la pièce. Notez également que par « joueur », nous faisons référence à un personnage joué par un acteur.

## Prétraitement des données

Pour commencer, vous devrez prétraiter les données que nous vous fournissons. Les données contenues dans le fichier CSV sont brutes, il est donc nécessaire de réorganiser certaines parties de celles-ci afin qu'elles puissent être correctement utilisées par la bibliothèque D3. Pour ce faire, vous devez compléter le fichier `scripts/preprocessing.js`.

Plus précisément, vous devrez effectuer ces étapes:

1. Mettez à jour les noms des joueurs dans les données pour que chaque mot commence par une majuscule suivi de lettres minuscules (fonction `cleanName`)
2. Implémentez une fonction pour trouver les noms des 5 joueurs avec le plus de des lignes prononcées dans la pièce (fonction `getTopPlayers`). Seulement ces joueurs seront directement affichés dans le diagramme à bandes groupées.
3. Regroupez et restructurez les données dans une structure imbriquée à être utilisée par D3 (fonction `summarizeLines`)
4. Modifiez la structure pour inclure une catégorie « *Other* », qui contient la somme des lignes dans cet acte qui sont prononcées par des joueurs n'appartenant pas au top 5 ayant le plus de lignes dans la pièce (fonction `replaceOthers`)

Pour vous aider à valider votre travail, la figure 1 illustre une partie de la structure de données résultante.

```
[
  {
    "Act": "1",
    "Players": [
      {
        "Player": "Benvolio",
        "Count": 34
      },
      {
        "Player": "Romeo",
        "Count": 50
      },
      {
        "Player": "Nurse",
        "Count": 20
      },
      {
        "Player": "Juliet",
        "Count": 16
      },
      {
        "Player": "Mercutio",
        "Count": 11
      },
      {
        "Player": "Other",
        "Count": 105
      }
    ]
  }
],
```

Figure 1: Extrait des données prétraitées

## Diagramme à bandes

Pour cette deuxième partie, vous devrez implémenter la partie principale de la visualisation de données. Tout d'abord, vous devrez mettre à jour les échelles utilisées pour positionner les éléments dans la visualisation. Deuxièmement, vous utiliserez ces échelles pour ajouter et positionner les groupes et sous-groupes au diagramme à bandes groupées. Pour ce faire, vous devez compléter le fichier `scripts/viz.js`.

Voici les étapes que vous devrez accomplir pour cette partie:

1. Définissez le domaine et l'image de l'échelle déterminant la position en x de chaque groupe de bandes (fonction `updateGroupScale`)
2. Définissez le domaine et l'image de l'échelle en y déterminant la longueur de chaque bande (fonction `updateYScale`)
3. Générez des groupes SVG qui contiendront chaque groupe de bandes (fonction `createGroups`)
4. Ajoutez des bandes dans chaque groupe avec la hauteur, la position et la couleur correspondante (fonction `drawBars`)

La figure 2 illustre ce à quoi le diagramme à bandes devrait ressembler après cette partie.

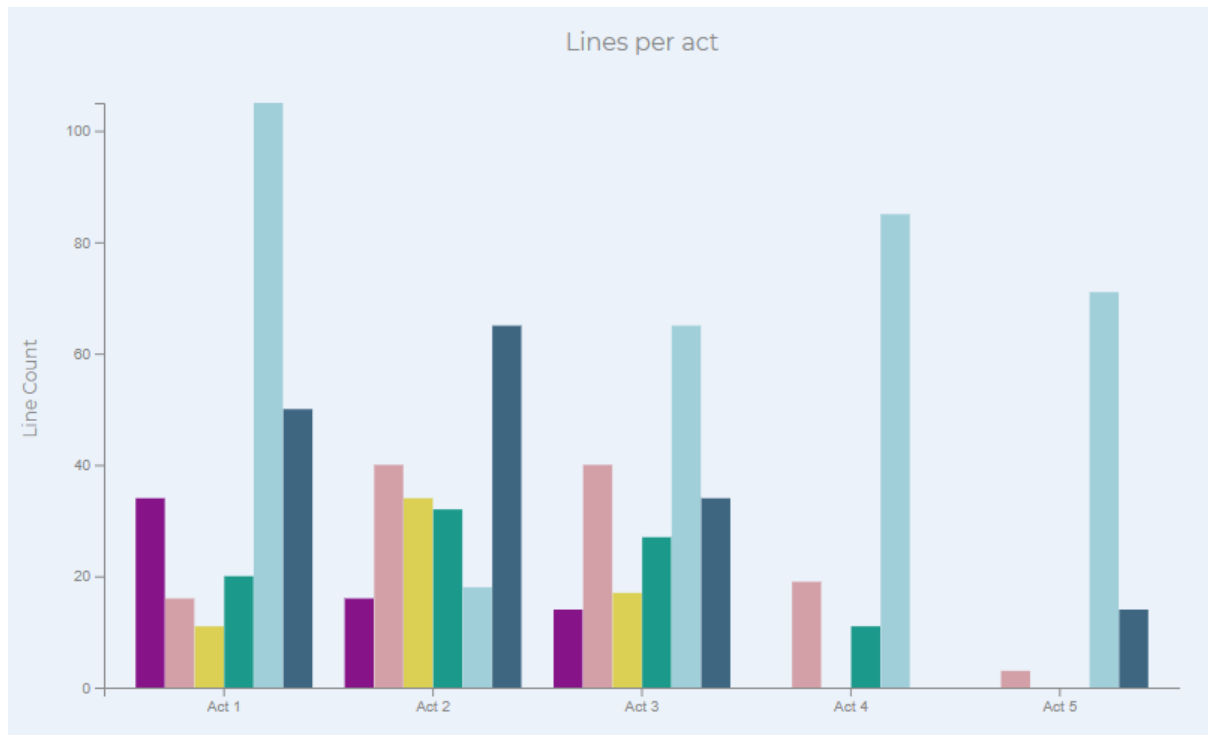


Figure 2: Le diagramme à bandes groupées

## Légende

Pour cette troisième partie, vous allez générer une légende dans le pied de page de la page Web. Prenez un moment pour remarquer dans les fichiers HTML et CSS fournis certains éléments pouvant être utile pour ajuster le positionnement et l'apparence visuelle de la légende. Pour compléter cette partie, vous devez remplir le fichier `scripts/legend.js`. L'intégralité du code de cette section peut être écrite dans la fonction `draw`. Assurez-vous que les couleurs de la légende correspondent à celles utilisées dans le graphique à bande pour chaque joueur.

La légende qui en résulte doit ressembler à celle de la figure 3.



Figure 3: La légende dans le pied de page

## Info-bulle

Pour cette quatrième partie, vous allez générer une info-bulle qui apparaît au-dessus chaque bande lorsqu'elle est survolée par le curseur. Cette info-bulle résume les informations associées à cette bande. L'info-bulle doit contenir comme titre l'acte associé au groupe de la bande, ainsi que le joueur et le nombre de lignes associés. Le titre doit être stylisé comme spécifié dans les commentaires du code. Pour compléter cette partie, vous devez compléter le fichier `scripts/tooltip.js`. L'intégralité du code de cette section peut être écrite dans la fonction `getContents`. Assurez-vous d'avoir également géré l'apparition et la disparition de l'info-bulle lors de la création des bandes du graphique.

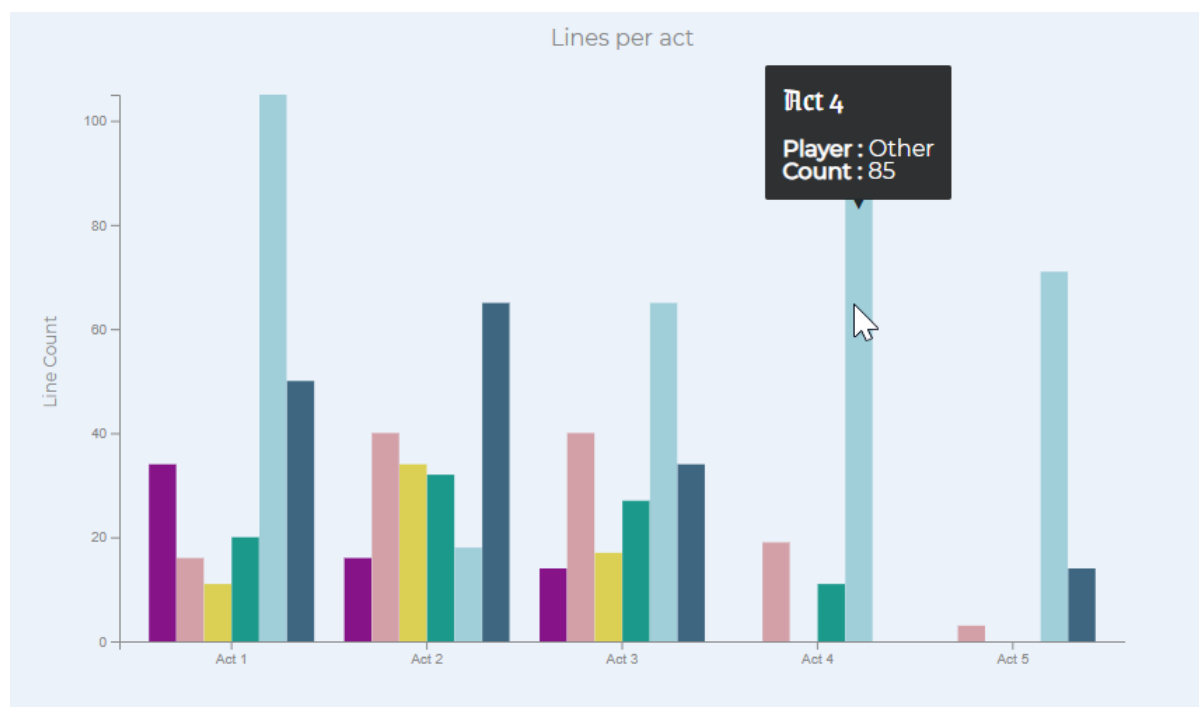


Figure 4: Le diagramme à bandes groupées avec une info-bulle au survol

Dans la figure 4, vous pouvez voir l'apparence de l'info-bulle lorsqu'une bande est survolée par le curseur.

## Soumission

Les instructions pour la soumission sont:

1. Vous devez placer le code de votre projet dans un fichier ZIP compressé nommé [matricule1\_matricule2\_matricule3.zip]
2. Le laboratoire doit être soumis avant le [22 mai 23h59m]

## Évaluation

Dans l'ensemble, votre travail sera évalué selon la grille suivante. Chaque section sera évaluée sur l'exactitude et la qualité du travail.

Exigence	Points
Prétraitement des données	6
Diagramme à bandes	8
Légende	3
Info-bulle	2
Qualité globale et clarté de la soumission	1
<b>Total</b>	<b>20</b>

# Références

---

[1] L. Larsen, "Shakespeare plays," Kaggle. Available: <https://www.kaggle.com/kingburrito666/shakespeare-plays>. [Accessed 30 07 2020].